UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/664,384	09/17/2003	Victor Roberts Augsburg	RPS920030054US1	4418

39698 7590 12/13/2006

DUKE W. YEE
YEE & ASSOCIATES, P.C.
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
LI, AIMEE J	
ART UNIT	**PAPER NUMBER**
2183	

DATE MAILED: 12/13/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/664,384	Applicant(s) AUGSBURG ET AL.
	Examiner Aimee J. Li	Art Unit 2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 29 September 2006.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-30 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-30 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-30 have been considered. Claims 8 and 15-16 have been amended as per Applicant's request.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Power of attorney as received on 11 July 2006 and Amendment as received on 29 September 2006.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 8-15 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 8's preamble recites "selectively the method comprising" which is unclear. As is, claim 8 suggests with "selectively" that certain steps in the method recited in the body of the claim. However, "the method comprising" language suggests that the method must at least comprise the following steps. Please clarify by either removing the "selectively" language or indicating which steps of the method are optional in the body of the claim.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 8-15 are rejected under 35 U.S.C. 102(b) as being taught by Hinton et al., U.S. Patent Number 5,555,432 (herein referred to as Hinton).

7. Referring to claim 8, Hinton has taught a single threaded or multi-threaded computer processor (Hinton column 1, lines 35-59), the method comprising:

a. Receiving sets of computer instructions in an instruction issue logic wherein each set of instructions comprises one or more instructions from each of one or more independent instruction threads (Hinton column 1, line 35 to column 2, line 11; column 3, lines 28-41; column 5, lines 23-29; and Figure 2);

b. Predicting a stage, within a multi-stage pipeline of the processor, in which a result from each instruction will be available (Hinton column 3, lines 41-53; column 7, lines 34-50; column 9, lines 22-40; column 11, lines 26-49; column 14, lines 42-64; Figure 2; Figure 7; and Figure 11). In regards to Hinton, Hinton predicts availability two cycles ahead of when the data will be available for use, but before the result is written back to memory and the instruction is retired. Therefore, Hinton predicts that the instruction results will be available in the stage two cycles from that point.

c. Identifying as dependent instructions those received instructions that require a result from a prerequisite instruction (Hinton column 1, line 59 to column 2, line 11; column 3, lines 31-40; column 5, line 66 to column 6, line 53; and Figure 2);

d. Determining a confidence factor for each instruction that indicates a probability that the instruction will complete all stages of the multi-stage pipeline without causing a stall (Hinton column 3, lines 41-53; column 7, lines 34-50; column 9,

lines 22-40; column 11, lines 26-49; column 14, lines 42-64; Figure 2; Figure 7; and Figure 11);

e. Determining said confidence factor for a given dependent instruction by calculating a critical distance between said given dependent instruction and its corresponding prerequisite instruction, wherein said critical distance is the number of stages between a stage when said given dependent instruction will need a result provided by said corresponding prerequisite instruction, and a stage when said provided result will be available (Hinton column 3, lines 41-53; column 7, lines 34-50; column 9, lines 22-40; column 11, lines 26-49; column 14, lines 42-64; Figure 2; Figure 7; and Figure 11); and

f. Issuing, from the instruction issue logic, instructions whose confidence factors are above a predetermined threshold (Hinton column 3, lines 54-62; column 7, lines 47-50; column 8, line 17 to column 9, line 50; column 12, lines 14-28; column 15, lines 26-35; column 20, line 63 to column 21, line 16; Figure 2; Figure 7; and Figure 11).

8. Referring to claim 9, Hinton has taught the method of claim 8, further comprising the steps of:

a. Storing the predicted pipeline stage for each instruction (Hinton column 23, line 39 to column 24, line 3); and,

b. Dynamically updating the stored predicted pipeline stage for each instruction based on a current contents of the pipeline available (Hinton column 3, lines 41-

53; column 7, lines 34-50; column 9, lines 22-40; column 11, lines 26-49; column 14, lines 42-64; Figure 2; Figure 7; and Figure 11);.

9. Referring to claim 10, Hinton has taught the method of claim 9, wherein the confidence factor for an instruction is determined based upon a current location and the predicted stage of the prerequisite instruction available (Hinton column 3, lines 41-53; column 7, lines 34-50; column 9, lines 22-40; column 11, lines 26-49; column 14, lines 42-64; Figure 2; Figure 7; and Figure 11).

10. Referring to claim 11, Hinton has taught the method of claim 10, further comprising the step of:

a. Dynamically recalculating the confidence factor for each instruction based on the current contents of the pipeline available (Hinton column 3, lines 41-53; column 7, lines 34-50; column 9, lines 22-40; column 11, lines 26-49; column 14, lines 42-64; Figure 2; Figure 7; and Figure 11).

11. Referring to claim 12, Hinton has taught the method of claim 9, further comprising the step of:

a. Identifying as dependent instructions those received instructions that require an operand from a memory of a computer system in which the computer processor operates instruction (Hinton column 1, line 59 to column 2, line 11; column 3, lines 31-40; column 5, line 66 to column 6, line 53; and Figure 2).

12. Referring to claim 13, Hinton has taught the method of claim 12, wherein the confidence factor for a dependent instruction is determined based upon a current location and the predicted stage of any prerequisite instruction and upon a probability that any required operand will be

found in a cache memory of the processor (Hinton column 3, lines 54-62; column 7, lines 47-50; column 8, line 17 to column 9, line 50; column 12, lines 14-28; column 15, lines 26-35; column 20, line 63 to column 21, line 16; Figure 2; Figure 7; and Figure 11).

13. Referring to claim 14, Hinton has taught the method of claim 13, further comprising the step of:

a. Dynamically recalculating the confidence factor for each instruction based on the current contents of the pipeline and a current contents of the processor cache memory (Hinton column 3, lines 41-53; column 7, lines 34-50; column 9, lines 22-40; column 11, lines 26-49; column 14, lines 42-64; Figure 2; Figure 7; and Figure 11).

14. Referring to claim 15, Hinton has taught the method of claim 8, wherein one or more instructions are issued from the instruction issue logic at each clock cycle (Hinton column 3, lines 54-62; column 7, lines 47-50; column 8, line 17 to column 9, line 50; column 12, lines 14-28; column 15, lines 26-35; column 20, line 63 to column 21, line 16; Figure 2; Figure 7; and Figure 11).

Claim Rejections - 35 USC § 103

15. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

16. Claims 1-7 and 24-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Swanson et al.'s "An Evaluation of Speculative Instruction Execution on Simultaneous

Multithreaded Processors" ©August 2003 (herein referred to as Swanson) in view of Teruyama, U.S. Patent Application US2003/0182536 (herein referred to as Teruyama).

17. Referring to claims 1, 8, 16, and 24, taking claim 16 as exemplary, Swanson has taught a single threaded computer processor with speculative instruction issue that increases throughput (Swanson page 315, paragraph 2). Swanson states on page 315, paragraph 2 that a simultaneous multithreading system can execute one thread and that all resources are dedicated to that thread, meaning that an SMT system can be a single threaded system with all buffers containing instructions for that thread. The computer processor comprising:

a. Multiple input buffers for receiving instructions from a thread of instructions (Swanson page 315, paragraphs 2-3; page 316, paragraph 2; page 319, paragraph 4; page 330, paragraph 1; and page 336, paragraph 4 to page 337, paragraph 1). In regards to Swanson, the individual buffers for each thread is inherent. Please see Parady, U.S. Patent Number 5,933,627 and Loikkanen and Bagherzadeh's "A Fine-Grain Multithreading Superscalar Architecture" ©1996 for more information.

b. Instruction issue logic that is connected to the independent input buffers (Swanson page 315, paragraphs 2-3; page 316, paragraph 2; page 319, paragraph 4; page 330, paragraph 1; and page 336, paragraph 4 to page 337, paragraph 1), wherein the instruction issue logic:

i. Receives instructions from the input buffers (Swanson page 315, paragraphs 2-3; page 316, paragraph 2; page 319, paragraph 4; page 330, paragraph 1; and page 336, paragraph 4 to page 337, paragraph 1);

ii. Determines a confidence factor for each instruction that indicates a probability that the instruction will complete all stages of the multi-stage pipeline without causing a stall (Swanson page 317, paragraph 1; page 326, paragraphs 4-7; page 327, paragraph 1 to page 328, paragraph 1; and page 336, paragraph 4 to page 337, paragraph 1); and,

iii. Issues instructions with confidence factors above a predetermined threshold (Swanson page 317, paragraph 1; page 326, paragraphs 4-7; page 327, paragraph 1 to page 328, paragraph 1; and page 336, paragraph 4 to page 337, paragraph 1).

18. Swanson has not taught

a. Predicts a stage, within a multi-stage pipeline of the processor, in which a result from each instruction will be available;

b. Identifies as dependent instructions those received instructions that require a result from a prerequisite instruction; and

c. Wherein a first stage of the multi-stage pipeline is connected to an output buffer of the instruction issue logic.

19. Teruyama has taught

a. Predicts a stage, within a multi-stage pipeline of the processor, in which a result from each instruction will be available (Teruyama paragraphs 0011-0014, 0032-0037, and 0082-0083; and Figure 5);

b. Identifies as dependent instructions those received instructions that require a result from a prerequisite instruction (Teruyama paragraphs 0011-0014, 0032-0037, and 0082-0083; and Figure 5); and

c. Wherein a first stage of the multi-stage pipeline is connected to an output buffer of the instruction issue logic (Teruyama paragraphs 0011-0014, 0032-0037, and 0082-0083; and Figure 5).

20. A person of ordinary skill in the art at the time the invention was made, and as taught by Teruyama, would have recognized that tracking the dependencies of an instruction ensures that instructions without dependencies are not cancelled inappropriately that results in reduced efficiency and speed (Teruyama paragraph 0013), thereby improving efficiency and speed of the processor. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the dependency tracker of Teruyama in the device of Swanson to improve processor efficiency and speed.

21. Claims 1 is functionally equivalent to claim 24 and is rejected for similar reasons. Claim 1 differs in that it is for a multithreaded system. Swanson teaches a simultaneous multithreaded system on, for example, page 315, paragraphs 2-3, and throughout the paper.

22. Referring to claims 2 and 25, taking claim 17 as exemplary, Swanson in view of Teruyama has taught the computer processor of claim 16, wherein the instruction issue logic stores the predicted pipeline stage for each instruction (Teruyama paragraphs 0011-0014, 0032-0037, and 0082-0083; and Figure 5) and, dynamically updates the stored predicted pipeline stage for each instruction based on a current contents of the pipeline (Swanson page 317, paragraph 1;

page 326, paragraphs 4-7; page 327, paragraph 1 to page 328, paragraph 1; and page 336, paragraph 4 to page 337, paragraph 1).

23. Referring to claims 3 and 26, taking claim 18 as exemplary, Swanson in view of Teruyama has taught the computer processor of claim 16, wherein the confidence factor for an instruction is determined based upon a current location and the predicted stage of the prerequisite instruction (Swanson page 317, paragraph 1; page 326, paragraphs 4-7; page 327, paragraph 1 to page 328, paragraph 1; and page 336, paragraph 4 to page 337, paragraph 1).

24. Referring to claims 4 and 27, taking claim 19 as exemplary, Swanson in view of Teruyama has taught the computer processor of claim 17, wherein the instruction issue logic dynamically recalculates the confidence factor for each instruction based on the current contents of the pipeline (Swanson page 317, paragraph 1; page 326, paragraphs 4-7; page 327, paragraph 1 to page 328, paragraph 1; and page 336, paragraph 4 to page 337, paragraph 1).

25. Referring to claims 5 and 28, taking claim 21 as exemplary, Swanson in view of Teruyama has taught the computer processor of claim 16, wherein the instruction issue logic further identifies as dependent instructions those received instructions that have a conflict over a shared resource within a computer system in which the computer processor operates (Teruyama paragraphs 0011-0014, 0032-0037, and 0082-0083; and Figure 5).

26. Referring to claims 6 and 29, taking claim 22 as exemplary, Swanson in view of Teruyama has taught the computer processor of claim 21, wherein the confidence factor for a dependent instruction is determined based upon a current location and the predicted stage of any prerequisite instruction and upon a predicted resolution of any identified shared resource conflict

(Swanson page 317, paragraph 1; page 326, paragraphs 4-7; page 327, paragraph 1 to page 328, paragraph 1; and page 336, paragraph 4 to page 337, paragraph 1).

27. Referring to claims 7 and 30, taking claim 23 as exemplary, Swanson in view of Teruyama has taught the computer processor of claims 22, wherein the instruction issue logic dynamically recalculates the confidence factor for each instruction based on a current contents of the pipeline and a current status of any shared resources (Swanson page 317, paragraph 1; page 326, paragraphs 4-7; page 327, paragraph 1 to page 328, paragraph 1; and page 336, paragraph 4 to page 337, paragraph 1).

28. Claims 16-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hinton et al., U.S. Patent Number 5,555,432 (herein referred to as Hinton) in view of Swanson et al.'s "An Evaluation of Speculative Instruction Execution on Simultaneous Multithreaded Processors" ©August 2003 (herein referred to as Swanson).

29. Referring to claim 16, Hinton has taught a computer processor with speculative instruction issue that increases throughput, the computer processor comprising:

a. Instruction issue logic that is connected to the independent input buffer (Hinton column 1, line 35 to column 2, line 11; column 3, lines 28-41; column 5, lines 23-29; and Figure 2), wherein the instruction issue logic:

i. Receives instructions (Hinton column 1, line 35 to column 2, line 11; column 3, lines 28-41; column 5, lines 23-29; and Figure 2);

ii. Predicts a stage, within a multi-stage pipeline of the processor, in which a result from each instruction will be available (Hinton column 3, lines 41-53; column 7, lines 34-50; column 9, lines 22-40; column 11, lines 26-49;

column 14, lines 42-64; Figure 2; Figure 7; and Figure 11). In regards to Hinton, Hinton predicts availability two cycles ahead of when the data will be available for use, but before the result is written back to memory and the instruction is retired. Therefore, Hinton predicts that the instruction results will be available in the stage two cycles from that point.

iii. Identifies as dependent instructions those received instructions that require a result from a prerequisite instruction (Hinton column 1, line 59 to column 2, line 11; column 3, lines 31-40; column 5, line 66 to column 6, line 53; and Figure 2);

iv. Determines a confidence factor for each instruction that indicates a probability that the instruction will complete all stages of the multi-stage pipeline without causing a stall (Hinton column 3, lines 41-53; column 7, lines 34-50; column 9, lines 22-40; column 11, lines 26-49; column 14, lines 42-64; Figure 2; Figure 7; and Figure 11); and,

v. Determines said confidence factor for a given dependent instruction by calculating a critical distance between said given dependent instruction and its corresponding prerequisite instruction, wherein said critical distance is the number of stages between a stage when said given dependent instruction will need a result provided by said corresponding prerequisite instruction, and a stage when said provided result will be available (Hinton column 3, lines 41-53; column 7, lines 34-50; column 9,

lines 22-40; column 11, lines 26-49; column 14, lines 42-64; Figure 2; Figure 7; and Figure 11); and

vi. Issues instructions with confidence factors above a predetermined threshold (Hinton column 3, lines 54-62; column 7, lines 47-50; column 8, line 17 to column 9, line 50; column 12, lines 14-28; column 15, lines 26-35; column 20, line 63 to column 21, line 16; Figure 2; Figure 7; and Figure 11); and

vii. Wherein a first stage of the multi-stage pipeline is connected to an output buffer of the instruction issue logic (Hinton Figure 1 and Figure 4).

30. Hinton has not taught a simultaneous multithreaded computer processor comprising multiple independent input buffers, wherein one set of buffers is provided for each of a plurality of independent threads of instructions. Swanson has taught a simultaneous multithreaded computer processor comprising multiple independent input buffers, wherein one set of buffers is provided for each of a plurality of independent threads of instruction (Swanson page 315, paragraphs 2-3; page 316, paragraph 2; page 319, paragraph 4; page 330, paragraph 1; and page 336, paragraph 4 to page 337, paragraph 1). In regards to Swanson, the individual buffers for each thread is inherent. Please see Parady, U.S. Patent Number 5,933,627 and Loikkanen and Bagherzadeh's "A Fine-Grain Multithreading Superscalar Architecture" ©1996 for more information. A person of ordinary skill in the art at the time the invention was made would have recognized, and as taught by Swanson, that simultaneous multi-threading increases the number of instructions executed, thereby improving hardware utilization (Swanson page 315, paragraph 2)

31. Referring to claim 17, Hinton in view of Swanson has taught the computer processor of claim 16, wherein the instruction issue logic stores the predicted pipeline stage for each instruction (Hinton column 23, line 39 to column 24, line 3) and, dynamically updates the stored predicted pipeline stage for each instruction based on a current contents of the pipeline (Hinton column 3, lines 41-53; column 7, lines 34-50; column 9, lines 22-40; column 11, lines 26-49; column 14, lines 42-64; Figure 2; Figure 7; and Figure 11).

32. Referring to claim 18, Hinton in view of Swanson has taught the computer processor of claim 16, wherein the confidence factor for an instruction is determined based upon a current location and the predicted stage of the prerequisite instruction (Hinton column 3, lines 41-53; column 7, lines 34-50; column 9, lines 22-40; column 11, lines 26-49; column 14, lines 42-64; Figure 2; Figure 7; and Figure 11).

33. Referring to claim 19, Hinton in view of Swanson has taught the computer processor of claim 17, wherein the instruction issue logic dynamically recalculates the confidence factor for each instruction based on the current contents of the pipeline (Hinton column 3, lines 41-53; column 7, lines 34-50; column 9, lines 22-40; column 11, lines 26-49; column 14, lines 42-64; Figure 2; Figure 7; and Figure 11).

34. Referring to claim 20, Hinton in view of Swanson has taught the computer processor of claim 16, wherein one or more instruction(s) is(are) issued from the instruction issue logic every clock cycle (Hinton column 3, lines 54-62; column 7, lines 47-50; column 8, line 17 to column 9, line 50; column 12, lines 14-28; column 15, lines 26-35; column 20, line 63 to column 21, line 16; Figure 2; Figure 7; and Figure 11).

35. Referring to claim 21, Hinton in view of Swanson has taught the computer processor of claim 16, wherein the instruction issue logic further identifies as dependent instructions those received instructions that have a conflict over a shared resource within a computer system in which the computer processor operates (Hinton column 1, line 59 to column 2, line 11; column 3, lines 31-40; column 5, line 66 to column 6, line 53; and Figure 2).

36. Referring to claim 22, Hinton in view of Swanson has taught the computer processor of claim 21, wherein the confidence factor for a dependent instruction is determined based upon a current location and the predicted stage of any prerequisite instruction and upon a predicted resolution of any identified shared resource conflict (Hinton column 3, lines 54-62; column 7, lines 47-50; column 8, line 17 to column 9, line 50; column 12, lines 14-28; column 15, lines 26-35; column 20, line 63 to column 21, line 16; Figure 2; Figure 7; and Figure 11).

37. Referring to claim 23, Hinton in view of Swanson has taught the computer processor of claims 22, wherein the instruction issue logic dynamically recalculates the confidence factor for each instruction based on a current contents of the pipeline and a current status of any shared resources (Hinton column 3, lines 41-53; column 7, lines 34-50; column 9, lines 22-40; column 11, lines 26-49; column 14, lines 42-64; Figure 2; Figure 7; and Figure 11).

Response to Arguments

38. Applicant's arguments with respect to claims 8-23 have been considered but are moot in view of the new ground(s) of rejection.

39. Applicant's arguments filed 29 September 2006, with regards to claims 1-7 and 24-30, have been fully considered but they are not persuasive.

40. Applicants argue in essence on pages 13-22

> ...it is clear Teruyama eschews or opposes Applicants' Feature(1), that is, predicting a stage in an instruction pipeline where results of respective instructions will be available...

41. This has not been found persuasive. Teruyama has taught in paragraphs 0011 that instructions dependent on a load instruction are issued speculatively when the probability is high that the data of the load instruction is in a cache, e.g. there is a high confidence for a cache hit. This means that Teruyama is predicting, based upon this confidence value, that the load instruction's data will be available at a later stage in the pipeline, such as the write-back stages W, X, Y, and Z and complete stage in Figure 2, when it is needed by the dependent instructions. Hence, the load instruction's data will have been fetched from the data cache and ready for use, e.g. it's A stage operation done, by the time the dependent instruction requires the data. There is nothing in Teruyama that states he is not predicting the stage the data is available in. To the contrary, Teruyama has taught predicting a stage the results of the respective instructions will be available, since he is predicting the load instruction results will be ready for use, e.g. the data has been loaded, by the time the data is required by the dependent instructions.

42. In addition, Applicant's argue in essence

> ...Teruyama proceeds to issue and execute successive instructions, until a cache miss occurs. This event then triggers operation of Teruyama's DLC 16, to detect dependencies between instructions being executed and the load instructions in the pipeline.

43. This has not been found persuasive. Teruyama states in paragraph 0011 that dependent instructions are issued only when it is predicted there is a high chance for a cache hit. Teruyama

describes in more detail how he handles a cache miss, because his invention is with regards to recovering from a miss-predicted cache hit. This is not to say that Teruyama arbitrarily executes instructions without regard to the probability of a cache hit for a load instruction. Also, Teruyama teaches in paragraphs 0082-0083 that the DLC is connected to <u>each entry in the instruction window buffer</u> and <u>updates various type of status bits of the instructions stored in the instruction window buffer</u>. He further says that "The DLC 16 retrieves an instruction depending on the load instruction in accordance with a cache miss signal…" not that the cache miss triggers detection, e.g. determining, dependent instructions. In order to retrieve dependent instructions, they had to have already been determined.

44. Applicants argue in essence on page 18

> …<u>nowhere</u> does Swanson appear to teach a <u>mechanism</u> for issuing instructions. that is equivalent to the instruction issue logic of Feature (2) of Claim 1.
>
> …In reciting the issuing of instructions, Feature (2) requires that instructions be <u>directed</u> into a pipeline processor from the instruction issue logic…Teruyama discloses <u>no</u> components or structures that <u>precede</u> instruction fetch unit 11…Circuits discussed in Teruyama, such as DLC 16, are all contained <u>within</u> the pipeline, and are thus not available to issue instructions <u>to</u> the pipeline…

45. This has not been found persuasive. First, the claim language of claim 1 states "issuing, from the instruction issue logic, instructions with confidence factors above a predetermined threshold." There is nothing in claim 1 reflecting that the issue logic issuing the instructions is not considered part of the pipeline. Claim 24 states this a little more clearly with "wherein a first stage of the multi-stage pipeline is connected to an output buffer of the instruction issue logic."

However, the claim language does not explicitly state where it considered the beginning of the pipeline stage to be. The fetch stage can be considered the beginning of a pipeline, as in Teruyama's drawings, but other pieces of art, such as Hinton U.S. Patents 5,555,432; 5,809,325; and 5,842,036, show that stage 1 of a pipeline, e.g. the beginning of a pipeline is the first stage in the execution unit, e.g. the first stage in the units of Teruyama's A stage in Figure 1. If we say that a pipeline does not "begin" until the beginning of the A stage, Teruyama would still be a multi-stage pipeline, since Figure 2 shows there are multiple stages after the A stage in Teruyama. Also, a pipeline commonly starts with a fetch stage, as shown in Teruyama, and there are inherently elements between the stages to buffer the outputs of each stage. Please see Heuring and Jordan's Computer Systems Design and Architecture ©1997 page 200, Modifications to Buses and the Data Path; page 201, Progressing Stage by Stage; Figure 5.2; and Figure 5.6, which show pipeline registers or latches between each stage to temporarily hold the input and output of each stage until the next stage is ready. Rosenberg's Dictionary of Computers, Information Processing, and Telecommunications Second Edition ©1987 shows under the fifth definition of "buffer" that a buffer is "a portion of storage for temporarily holding input or output data."

46. Applicants argue in essence on pages 18-20

> ...One of skill in the art therefore would not be motivated to undertake any such modification, without clear guidance to do so that was shown by the prior art. As discussed above, Applicants have been provided with no such guidance or motivation.

> ...Teruyama is not concerned with speculative processing of instructions, and thus effectively teaches away from any combination with Swanson, or other reference, to realize key elements...

47. This has not been found persuasive. The amount of modification needed for a primary reference to meet the claim limitations is not a non-motivating factor. A non-motivating factor would be if the primary reference explicitly stated that it cannot perform the missing functions as it would be contrary to the primary reference's purpose and/or prevent the primary reference from functioning properly. This argument seems to suggest that Applicant is literally trying to incorporate the device taught in the secondary reference, i.e. bodily incorporate. Combination rejections are not based upon literally incorporating the device taught in the secondary reference into the primary reference, but what the two together would have suggested to a person of ordinary skill in the art. The test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference; nor is it that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981). Teruyama explicitly states in paragraph 0013 that unnecessarily canceling non-dependent instructions following a cache miss load instruction deteriorates processor efficiency, and that his device attempts to solve this inefficiency by tracking dependent instructions for speculatively executed instructions (Teruyama, paragraph 0014). Teruyama further states in 0014 that his device is for a system the "speculatively issues instructions out-of-order", so he does not teach away Swanson, as is argued.

Conclusion

48. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Hinton et al., U.S. Patent Numbers 5,809,325 and 5,842,036, have taught scheduling instructions for issue two cycles ahead of when at least one of its operands will be available.

b. Call et al., U.S. Patent Number 5,848,256, has taught a system which schedules instructions according to their instruction dependency.

c. Hicks et al., U.S. Patent Number 5,864,341, has taught a system which schedules instructions and bypasses operands from previous instructions.

d. Walker, U.S. Patent Number 5,870,580, has taught tracking instruction dependencies in an out-of-order execution system and forwarding results to dependent instructions.

49. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

50. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

51. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J. Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:00am-4:30pm.

52. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

53. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

AJL
Aimee J. Li
7 December 2006

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100